Job No.: 228-124893
Ref.: JP 2000/032426/RCA89615_JP_JP2000032426/JMF(Jill)/Order No. ART25
Translated from Japanese by the McElroy Translation Company
800-531-9977 customerservice@mcelroytranslation.com

(19) JAPANESE PATENT OFFICE (JP)
(12) KOKAI TOKUHYO PATENT GAZETTE (A)
(11) PATENT APPLICATION PUBLICATION No. P2000-32426A
(43) Publication Date January 28, 2000

(51) Int. Cl.[7]:	Identification Codes:	FI		Theme Codes (Reference):
H 04 N 7/16		H 04 N 7/16	Z	5C064
H 04 L 12/56		H 04 L 11/20	102Z	5K030

Examination Request: Not filed
No. of Claims: 5 (Total of 9 pages; OL)

(21) Filing No.: Hei 10[1998]-192083
(22) Filing Date: July 7, 1998

(71) Applicant: 000004075
Yamaha Corporation
10-1 Nakazawa-cho
Hamamatsu-shi, Shizuoka-ken

(72) Inventor: Shigeki Fujii
Yamaha Corporation
10-1 Nakazawa-cho
Hamamatsu-shi, Shizuoka-ken

(74) Agent: 100092820
Katsu Itami, Patent Attorney

F Terms (Reference):
5C064 BA01 BB05 BC10 BC13 BC20
BD08 BD09 BD14
5K030 GA12 HB02 HB28 JA01 JL01
JT04 LD07 MB13

(54) [Title] TRANSPORT STREAM RECEIVER

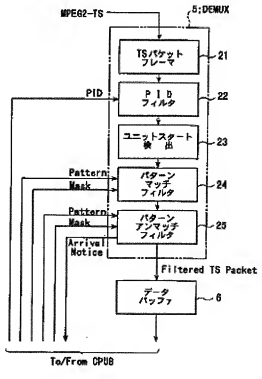
(57) Abstract

Problem

To achieve robust TS demultiplex processing even when a transmission fault or a reception fault occurs.

Means to solve the problem

A desired packet containing an ID specifying that this is the packet to be acquired and containing a version number that is changed whenever the contents of said packet is changed is acquired from an MPEG2-TS whenever the version number changes. Then, a pattern matching filter 24 extracts that packet when there is a pattern match for the ID from the TS. When a pattern non-match filter 25 detects that the version number contained in the packet extracted by filter 24 differs from the version number currently being monitored, it extracts that packet. When a desired packet is extracted by this filter 25, a CPU 8 acquires this packet and uses the newly detected version number to update the version number being monitored.



Key: 6 Data buffer
21 TS packet frame
22 PID filter
23 Unit start detection
24 Pattern matching filter
25 Pattern non-matching filter

[There are no amendments to this patent.]

Claims

1. Transport stream receiver that acquires a desired packet containing first information specifying the packet to be acquired and also containing second information that is changed whenever the content of said packet is changed from a multiplexed transport stream, whenever said second information is changed,

characterized in that the transport stream receiver is equipped with: a first filter that detects the aforementioned first information in the aforementioned input transport stream and extracts the aforementioned desired packet;

a second filter that, when it detects that the aforementioned second information contained in the packet extracted by this first filter differs from the second information currently being monitored, extracts the aforementioned packet that was extracted by the first filter;

and a control means that, when a desired packet is extracted by this second filter, acquires said packet and uses the newly detected second information to update the second information to be monitored next.

2. Transport stream receiver recorded in Claim 1, characterized in that the aforementioned desired packet is data that is transmitted repeatedly with the same contents during the period in which the aforementioned second information is updated.

3. Transport stream receiver recorded in Claim 1 or 2, characterized in that the aforementioned first filter extracts the relevant packet when a match is detected between a first pattern and a prescribed matching field of the aforementioned packet to be acquired.

4. Transport stream receiver recorded in any one of Claims 1-3, characterized in that the aforementioned second filter extracts the relevant packet when non-matching is detected between a second pattern and a prescribed matching field of the aforementioned packet to be acquired.

5. Transport stream receiver recorded in any one of Claims 1-4, characterized in that the aforementioned control means acquires the aforementioned desired packet periodically regardless of the updating of the aforementioned second information.

Detailed explanation of the invention

[0001]

Technical field of the invention

The present invention pertains to a receiver that acquires a desired packet from a multiplexed transport stream, such as an MPEG-2 (Moving Picture Experts Group Phase 2)—TS (Transport Stream); in particular, it pertains to a receiver with improved adaptability when a fault occurs.

[0002]

Prior art

With a digital data broadcast, like a multiplexed transmission stream—for example, with a digital CS broadcast—video and audio are digitized in an MPEG2-TS format and transmitted to a receiver. At this time, program-related information known as program specification information (PSI) and service information (SI) are transmitted simultaneously in the TS. All of this information is packetized into data of a fixed length of 188 bytes, known as a TS packet, and is multiplexed. A digital broadcast receiver first acquires the PSI, extracts the packet ID (PID) and the like of the video or audio stream, and while that information is being received this must be continually acquired and updated. The PSI and the SI comply with a section format that is defined by the MPEG2 system. In this section there is a field (5 bits) known as a 'version number' that is provided for the management of changes in the data. The method operation of this version number depends upon the broadcast service, but typically the version number is incremented due to the changing of the data. Accordingly, a digital broadcast receiver possesses information that holds a value wherein the version number is incremented by one after a given PSI is received.

[0003]

In addition, in order to extract only the desired TS packet from the multiplexed MPEG2-TS the receiver is provided with a DEMUX (demultiplexer). The DEMUX has a function whereby it specifies the PID and a subsequent, several-byte bit pattern and compares this with the input stream by means of an internal judgment circuit, and when there is a match with the specified pattern, that packet is written to a prescribed memory and the arrival of that data is reported to a host CPU and the like.

[0004]

Problem to be solved by the invention

With a conventional system there is no problem when transmission/reception are occurring normally; however, there is a problem when a transmission fault, reception fault or the like occurs and when the desired information is correct in terms of grammar and content but the expected bit pattern is transmitted with a different pattern. In this case, the receiver is unable to detect a pattern that matches the specified pattern, so a long waiting state—in some cases, of several days—ensues. In particular, there is a problem when version numbers are nonconsecutive.

[0005]

This problem will be explained in more detail with reference to Figure 7(a), where currently TS packets with version number = 1 are being acquired and there is preparation to acquire TS packets with version number = 2. For program information or the like, the version number is, for example, updated once every day. However, the receiving side does not know to begin receiving, so this type of TS packet is retransmitted repeatedly with identical contents at fixed intervals. The DEMUX of the receiver performs pattern matching with respect to a prescribed matching field that includes the packet ID and the version number; in this example, it waits for a packet with packet ID = 80 and version number = 2, ignoring packets with version number = 1. When a communications fault or the like occurs and, for example, a switching of the transmitting-side MUX (multiplexer) occurs, the version number becomes 4 and the consecutiveness of the version numbers is lost. In this case, the receiving side is waiting version number = 2, and thus is unable to acquire the updated packet, resulting in a long wait.

[0006]

To avoid such problems a method can be considered wherein data is acquired while the version number is ignored, but the PSI and similar information is frequently retransmitted, so for each retransmission the host CPU must judge whether the data has been refreshed, which is inefficient. Another method of avoidance is for the host CPU to check the data acquisition status periodically and to perform a PSI reacquisition process when it is determined that there is an error, but it is difficult to accurately determine when errors occur, so this cannot be considered a fundamental solution proposal.

[0007]

The present invention was devised in response to such problems, and the objective is to provide a transport stream receiver capable of achieving robust TS demultiplex processing even when a transmission fault or a reception fault occurs.

[0008]

Means to solve the problem

For a transport stream receiver that acquires a desired packet containing first information specifying the packet to be acquired and also containing second information that is changed whenever the contents of said packet is changed from a multiplexed transport stream, whenever the aforementioned second information is changed, the transport stream receiver according to the present invention is characterized in that it is equipped with: a first filter that detects the aforementioned first information in the aforementioned input transport stream and extracts the aforementioned desired packet; a second filter that, when it detects that the aforementioned second information included in the packet extracted by this first filter differs from the second information currently being monitored, extracts the aforementioned packet that was extracted by the first filter; and a control means that, when a desired packet is extracted by this second filter, acquires said packet and uses the newly detected second information to update the second information to be monitored next.

[0009]

Here, the aforementioned desired packet is data that is transmitted repeatedly with the same contents during the period in which the aforementioned second information is being updated. The first filter, for example, can be a filter that extracts the relevant packet when a match is detected between a first pattern and a prescribed matching field of the aforementioned packet to be acquired. Furthermore, the second filter can be a filter that extracts the relevant packet when non-matching is detected between a second pattern and a prescribed matching field of the aforementioned packet to be acquired.

[0010]

By means of the present invention, monitoring is not performed for a match between the second information that is updated whenever the contents of the packet are updated and the second information of the packet that is to be acquired next and that has been updated; instead, monitoring is performed for non-matching with the second information of the current packet. Therefore, when the second information is not updated in the determined sequence due to the occurrence of a fault, an update is detected by means of the non-matching with the current second information.

[0011]

In addition, if the control means are means that acquire the aforementioned desired packet periodically regardless of the updating of the aforementioned second information, it is

possible to some extent to avoid a long waiting state when second information that is identical to that of the previously received information is received occasionally.

[0012]

Embodiment of the invention

In the following, a preferred embodiment of the present invention will be explained with reference to the figures. Figure 1 is a block diagram showing the structure of a CS digital broadcast receiver according to one application example of the present invention. A CS reception signal received through an antenna 1 is demodulated by a tuner 2, and an error correction process is performed by an error correction unit 3. The error-corrected data is unscrambled by a descrambler 4 and supplied to a demultiplexer (DEMUX) 5. DEMUX 5 extracts only the required information from the multiplexed information; control information, program-related information and the like are supplied to a CPU 8 through a buffer 6 and a system bus 7, and video data and audio data are supplied to an MPEG A/V decoder 9 and decoded. The decoded video data is encoded as an NTSC video signal by a video encoder 10 and then is output. In addition to CPU 8, a switch/remote control unit 12 that receives infrared signals from a remote control 11, a RAM/ROM 13 that stores the CPU 8 operating program and that provides a work area, a conditional access module 14 that obtains a key used to perform the descrambling process, and a modem 15 that transmits and receives data to/from a telephone line network are connected to system bus 7.

[0013]

Figure 2 is a functional block diagram showing the structure of DEMUX 5 of the aforementioned system in more detail. It should be noted that only the flow of the control information and program-related information and the like are shown herein; the flow of the video data and audio data is omitted. This DEMUX 5 has a structure that includes a TS packet frame 21, a PID filter, a unit start detection unit 23, a pattern matching filter 24, and a pattern non-matching filter 25.

[0014]

Figure 3 shows examples of a typical MPEG2-TS packet and a section comprised thereof. As shown in (a) of the same figure, the TS packet is a fixed-length, 188-byte packet comprised of a 4-byte header and 184 bytes of data bytes. The header contains information such as a synchronization byte (8 bits), a payload unit start (1 bit), and a PID (13 bits). At TS packet frame 21 within DEMUX 5 the synchronization byte and the like are referenced and a process is performed to extract the 188-byte, fixed-length TS packet. The PID information shown in the TS

header of the TS packet that has been extracted by TS packet frame 21 is checked by PID filter 22, and a packet that matches the PID specified by CPU 8 is extracted. The data byte portion is extracted from the extracted TS packet, the data byte portions extracted from multiple TS packets are joined together, and as shown in Figure 3(b) a data block if formed with a section format defined by the MPEG2 system. A 1-bit payload_unit_start_indicator which indicates whether a section header is included in the data bytes of the packet is provided as the information which indicates which TS packet will be at the head when the section is formed.

[0015]

Figure 3(c) shows an example of the arrangement of the bits of a formed section. A section header is provided at the beginning of the section. In this example, the section header comprises 8 bytes and includes, for example, the following information.

① table_id (8 bits)

Table identification information

② section_length (12 bits)

Section length

③ table_id_extention [sic; extension]

A field used to expand the table_id

④ version_number

Information used to determine whether there is a table information update

⑤ section_number

An ID number when a table comprises multiple sections

⑥ last_section_number

The ID number of the final section when a table comprises multiple sections.

[0016]

At pattern matching filter 24, this section header is compared with a matching pattern [sic] supplied by CPU 8. CPU 8 also supplies pattern matching filter 24 with a mask pattern, and the matching process is executed with the bits of the section header that are unrelated to the pattern matching process and that are masked by this mask pattern. Here, as the first information used to specify the TS packet to be acquired, for example, if the settings are such that the process waits for table_id =23h, table_id_extention [sic; extension] = 0456h; and, as the second information that is changed whenever the contents of the packet are changed, the process waits for version_number ! = 1 (not equal to 1), then with the conventional pattern matching an update to the contents of the TS packet is detected by picking up all of the aforementioned first and second

information using a mask pattern as shown in Figure 3(e) and by detecting a match with the aforementioned pattern.

[0017]

By contrast, with the present invention an update to the contents of the TS packet is detected using two filters—a pattern matching filter (the first filter) 24 and a pattern non-matching filter (the second filter) 25. First, at pattern matching filter 24 only the table_id and the table_id_extension portions, which are the first information, are picked up using a mask pattern as shown in Figure 4(c), and the other portions are masked. Then, as shown in (b) of the same figure, pattern matching is performed with the process currently waiting for table_id = 23h and table_id_extension = 0456h, and when there is a match the packet is extracted.

[0018]

Next, at pattern non-matching filter 25, only the version_number, which is the second information, is picked up using a mask pattern as shown in Figure 4(e), and the other portions are masked. Then, as shown in (d) of the same figure, when non-matching is detected as a result of the pattern non-matching process that is currently waiting for version_number ! = 1, that packet is written to a data buffer (queue buffer memory) 6 connected to DEMUX 5, and when the section is completed, CPU 8 is notified that the desired data has arrived.

[0019]

As shown in Figure 7(b), by means of this device when TS packets with version_number = 1 are being acquired and there is a fault in the communications line or the like resulting in the version_number = 4, this TS packet can be acquired based on the condition that the version_number ! = 1 (not equal to 1).

[0020]

In addition, if data with the same version number arrives accidentally after a fault occurs, this cannot be detected even with the aforementioned device. Accordingly, CPU 8 can periodically execute a data pickup process regardless of the version number, and after the data is acquired the content of the data can be compared with the data acquired previously to determine whether it is new data. This process is equivalent to a periodic resetting of the data acquisition process.

[0021]

Figure 5 and Figure 6 are flowcharts for the purpose of explaining this process. As shown in Figure 5, with the normal process, whenever a notice is detected that data has arrived from DEMUX 5 (Arrival Notice), data (version_number = x) is read from data buffer 6 (S1), and the data acquisition setting for the next data (version_number ! = x) is made (S2). Then, the acquisition monitor timer is reset (S3). In addition, the acquisition monitoring process shown in Figure 6 is executed with the normal process in the background. In other words, until the acquisition reset time arrives (S12), the timer is incremented (S11) and the timer waits for one unit time (S13) repeatedly, and when the acquisition reset time arrives, the current wait for data is released and data acquisition is performed for the settings (S14) while the version is ignored. Next, the acquisition monitor timer is reset (S15). Thus, by applying an interrupt periodically the normal state will be restored after the acquisition monitor time has elapsed, even if the same version number is received occasionally after a fault occurs. However, if this acquisition monitor time is too short it will increase the load on CPU 8, and if it is too long the wait time until a return to the normal state after an error occurs will increase; however, the possibility of there being a fault and the same version number appearing simultaneously is considered very slight, so resetting once every several hours will be sufficient.

[0022]

Effect of the invention

As described above, by means of the present invention monitoring is not performed for a match between the second information that is updated whenever the contents of the packet are updated and the second information of the packet that is to be acquired next and that has been updated; instead, monitoring is performed for non-matching with the second information of the current packet. Therefore, when the second information is not updated in the determined sequence due to the occurrence of a fault, an update is detected by means of the non-matching with the current second information, and the adaptability with respect to the occurrence of a fault can be improved.

Brief description of the figures

Figure 1 is a block diagram showing the structure of a CS receiver according to one application example of the present invention.

Figure 2 is a functional block diagram of the DEMUX of said device.

Figure 3 is a diagram showing the structure of an MPEG2-TS packet and a section received by said device.

Figure 4 is a diagram for the purpose of explaining the pattern matching process and the pattern non-matching process of said device.

Figure 5 is a flowchart showing a typical process that further improves said device.

Figure 6 is a flowchart that similarly shows a data acquisition monitoring process.

Figure 7 is a diagram comparing the operation/effect of the present invention with an example of the prior art.

Explanation of codes

5 DEMUX
21 TS packet frame
22 PID filter
23 Unit start detection unit
24 Pattern matching filter
25 Pattern non-matching filter

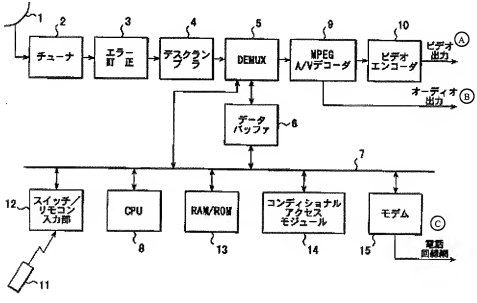


Figure 1

Key: A Video output
B Audio output
C Telephone line network
2 Tuner
3 Error correction
4 Descrambler

6 Data buffer
9 MPEG A/V decoder
10 Video encoder
14 Conditional access module
15 Modem

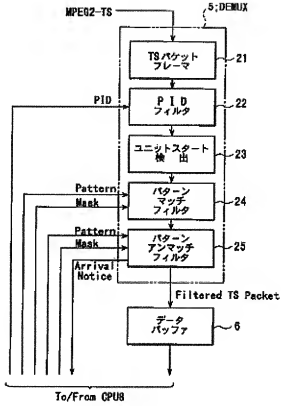


Figure 2

Key: 6 Data buffer
21 TS packet frame
22 PID filter
23 Unit start detection
24 Pattern matching filter
25 Pattern non-matching filter

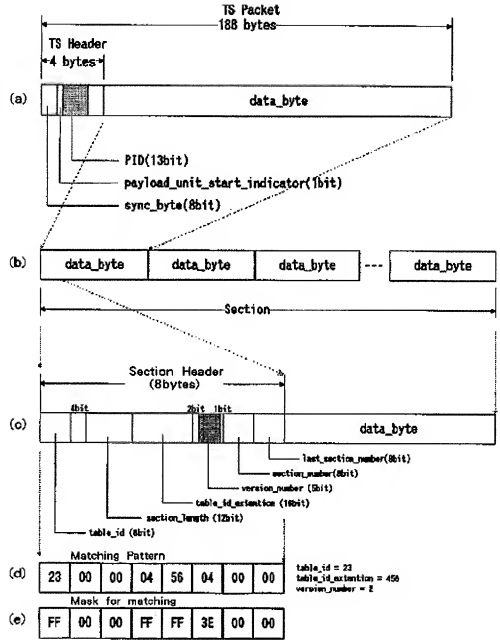

TS Packet
188 bytes
TS Header
4 bytes
(a)
data_byte
PID(13bit)
payload_unit_start_indicator(1bit)
sync_byte(8bit)
(b)
data_byte
data_byte
data_byte
data_byte
Section
Section Header
(8bytes)
4bit
2bit
1bit
(c)
data_byte
last_section_number(8bit)
section_number(8bit)
version_number (5bit)
table_id_extention (16bit)
section_length (12bit)
table_id (8bit)
Matching Pattern
(d)
23
00
00
04
56
04
00
00
table_id = 23
table_id_extention = 456
version_number = 2
Mask for matching
(e)
FF
00
00
FF
FF
3E
00
00

Figure 3

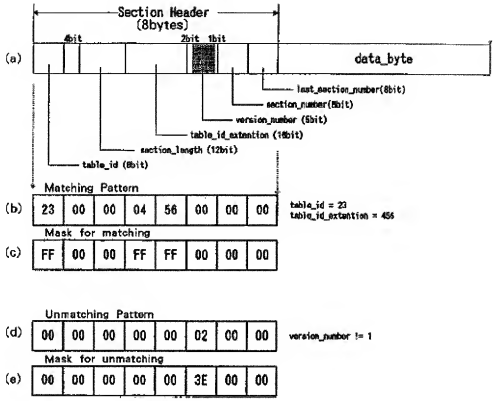

Section Header
(8bytes)
4bit
2bit
1bit
(a)
data_byte
last_section_number(8bit)
section_number(8bit)
version_number (5bit)
table_id_extention (16bit)
section_length (12bit)
table_id (8bit)
Matching Pattern
(b)
23 00 00 04 56 00 00 00
table_id = 23
table_id_extention = 456
Mask for matching
(c)
FF 00 00 FF FF 00 00 00
Unmatching Pattern
(d)
00 00 00 00 00 02 00 00
version_number != 1
Mask for unmatching
(e)
00 00 00 00 00 3E 00 00

Figure 4

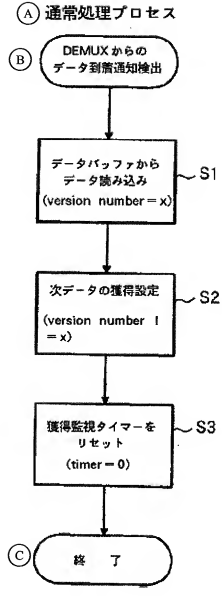


Figure 5

Key: A Normal processing
B Detection of notification of arrival of data from DEMUX
C End
S1 Read data from data buffer
S2 Set acquisition of next data
S3 Reset acquisition monitor timer

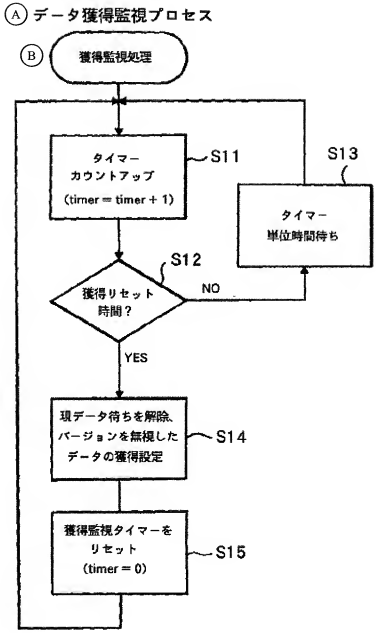


Figure 6

Key: A Data acquisition monitoring process
B Acquisition monitoring process
S11 Increase timer count
S12 Acquisition reset time?
S13 Wait for unit time of timer
S14 Release current data wait, set data acquisition while version is ignored
S15 Reset acquisition monitor timer

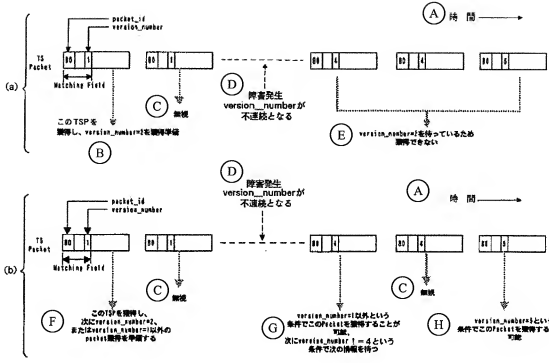


Figure 7

Key:
- A Time
- B Acquire this TSP, prepare for acquisition of [packet for which] version_number = 2
- C Ignore
- D Fault occurs; version_numbers become nonconsecutive
- E Waiting for version_number = 2, so acquisition cannot occur
- F Acquire this TSP, next prepare for acquisition of [packet for which] version_number = 2 or [for which] version_number = other than 1
- G This packet can be acquired with the condition that the version_number = other than 1; next, wait for next information with the condition that the version_number ! = 4
- H This packet can be acquired with the condition that the version_number = 5